AD-A092 951 TEXAS UNIV AT AUSTIN CENTER FOR CYBERNETIC STUDIES F/G 12/2

THE M-TRAVELLING SALESMEN PROBLEM: A DUALITY BASED BRANCH-AND-B--ETC(U)

AUG 80 I ALI, J KENNINGTON N00014-75-C-0569

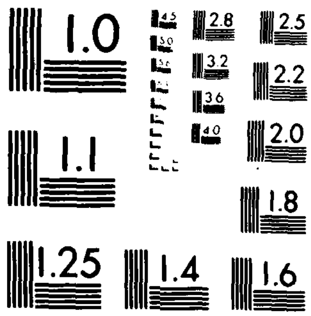
UNCLASSIFIED CCS-RR-382 NL

END

DATE FILMED

1 81

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

AD A092951

LEVEL II

Research Report CCS 382

THE M-TRAVELLING SALESMEN PROBLEM: A DUALITY BASED BRANCH-AND-BOUND ALGORITHM

by

I. Ali
J. Kennington*

CENTER FOR CYBERNETIC STUDIES
The University of Texas
Austin, Texas 78712

80 12 17 018

Research Report CCS 382

THE M-TRAVELLING SALESMEN PROBLEM: A DUALITY BASED BRANCH-AND-BOUND ALGORITHM

by

I. Ali
J. Kennington*

August 1980

*Southern Methodist University

This research was supported in part by the Air Force Office of Scientific Research under Contract Number AFOSR 77-3151, as part of the first author's dissertation, and by Office of Naval Research Contract N00014-75-C-0569 with the Center for Cybernetic Studies, The University of Texas at Austin.

CENTER FOR CYBERNETIC STUDIES

A. Charnes, Director
Business-Economics Building, 203E
The University of Texas at Austin
Austin, Texas 78712
(512) 471-1821

ABSTRACT

This paper presents a new model and branch-and-bound algorithm for the m-travelling salesmen problem. The algorithm uses a Lagrangean relaxation, a subgradient algorithm to solve the Lagrangean dual, a greedy algorithm for obtaining minimal m-trees, penalties to strengthen the lower bounds on candidate problems, and a new concept known as staged optimization. Computational experience for both symmetric and asymmetric problems having up to 100 cities is presented.

ACKNOWLEDGEMENT

This research was supported in part by the Air Force Office of Scientific Research under Contract Number AFOSR 77-3151. This work was part of the first author's dissertation.

1. OVERVIEW

This paper extends the highly successful algorithm of Held and Karp [11, 12] for the travelling salesman problem, to the m-travelling salesmen problem. The algorithm involves the use of a Lagrangean dual within a branch-and-bound structure. A subgradient procedure is used to solve the dual and it is shown that a greedy algorithm may be used to evaluate points of the dual function.

We now formally state the *m-travelling salesmen problem*. We define a *tour* beginning at city n as a sequence of distinct cities $\{n, i_1, i_2, \ldots, i_\ell\}$ where $\ell \geq 1$. Given n cities $(1, \ldots, n)$ and m salesmen, all based at city n, we wish to find a set of m tours such that each city other than n is a member of exactly one tour. Let c_{ij} denote the distance from city i to city j and let $c_{ni_1} + c_{i_1 i_2} + \ldots c_{i_{\ell-1} i_\ell} + c_{i_\ell n}$ denote the distance for the tour $\{n, i_1, \ldots, i_\ell\}$. The objective is to select m tours (one for each salesman) such that the total distance for all tours is a minimum. A related problem has also been discussed in the literature in which the objective is to select at most m tours with total minimum distance (see [2]). Clearly, for m = 1 both models are the classical travelling salesman problem. If the distances $c_{ij} = c_{ji}$ for all city pairs, then this model is called *symmetric*; otherwise, it is called *asymmetric*.

This model was first formulated in integer programming terms by Miller, Tucker, and Zemlin [17]. Svetska and Huckfeldt [18] developed a specialized branch-and-bound algorithm for this problem which uses a linear programming relaxation. Also, Gavish and Srikanth [7] developed

a branch-and-bound algorithm which uses a different relaxation. The Gavish-Srikanth model does not permit the use of a greedy algorithm to solve subproblems whereas our model does. Bellmore and Hong [5] proved that the asymmetric version of this model was equivalent to an asymmetric travelling salesman problem on $n + m - 1$ cities. The question of whether this problem should be attacked directly as in [7, 18] and the present work or whether it should be converted to its one salesman equivalent and solved using [3] or [12] is an open question.

II. THE MODEL

In this section we present a new model for the m-travelling salesmen problem. The model is developed using the notion of an m-tree, which is a generalization of the Held-Karp 1-tree. The m-tree is defined such that it has the matroidal property so that the integer programming relaxation is solvable via a greedy algorithm. The original generalization proposed by Held and Karp [11] is not matroidal.

Let the decision variable

$$x_{ij} = \begin{cases} 1, & \text{if some salesman travels from city } i \text{ to city } j; \\ 0, & \text{otherwise.} \end{cases}$$

Assuming that each salesman is based at city n, we call any tour on cities 1, ..., n - 1 a *subtour*. Then the m-travelling salesmen problem may be stated mathematically as follows:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{1}$$

$$\text{s.t.} \sum_{k=1}^{n} x_{kj} = \left\{ \begin{array}{l} 1, \text{ for } j = 1, \ldots, n-1 \\ m, \text{ for } j = n \end{array} \right\} \tag{2}$$

$$\sum_{k=1}^{n} x_{ik} = \left\{ \begin{array}{l} 1, \text{ for } i = 1, \ldots, n-1 \\ m, \text{ for } i = n \end{array} \right\} \tag{3}$$

$$\left. \begin{array}{lll} x_{ij} = & 0 \text{ or } 1 & (\text{all } i, j) \\ x_{ii} = & 0 & (\text{all } i) \end{array} \right\} \tag{4}$$

$$\text{no subtours.} \tag{5}$$

For the classical travelling salesman problem (i.e. $m = 1$), a pair of expository papers by Held and Karp [11, 12] showed that (4) and (5) could be replaced by a constraint that required the graphical structure associated with any feasible vector x to be a 1-tree. A *1-tree* on a graph having n nodes (cities) is a spanning tree on n-1 nodes and two distinct edges connecting node n to two other nodes. Letting $Y = \{(x_{11}, \ldots, x_{1n}, \ldots, x_{n1}, \ldots, x_{nn}) : x_{ii} = 0, x_{ij} = 0$ or 1 and the edges ij having $x_{ij} = 1$ form a 1-tree$\}$, the travelling salesman problem may be stated as (1), (2), (3), $x \in Y$, and $m = 1$.

We now generalize the notion of a 1-tree to an m-tree.

> *An m-tree on a graph having n nodes is an acyclic graph having n-m-1 edges on n-1 nodes and 2m edges such that the edge (n,i) never appears more than twice.*

Note that every m-tree has n+m-1 edges and an m-tree need not be connected. A set of 2-trees is illustrated in Figure 1 while Figure 2 illustrates some graphs having n+m-1, (6), edges which are not 2-trees. The graph in Figure 2a is not acyclic on nodes 1 and 2 and the graph in Figure 2b has three copies of the edge (5,2).

Letting $X \doteq \{(x_{11}, \ldots, x_{1n}, \ldots, x_{n1}, \ldots, x_{nn}): x_{ii} = 0$, $x_{ij} = 0$ or 1, and the edges ij with $x_{ij} = 1$ form an m-tree$\}$, the m-travelling salesmen problem may be stated as follows:

$$\min \sum_{ij} c_{ij}x_{ij} \tag{6}$$

$$\text{s.t.} \sum_{k} x_{kj} = \begin{cases} 1, \text{ for } j = 1, \ldots, n-1 \\ m, \text{ for } j = n \end{cases} \tag{7}$$

$$\sum_k x_{ik} = \begin{cases} 1, \text{ for } i = 1, \ldots, n-1 \\ m, \text{ for } i = n \end{cases} \tag{8}$$

$$x \in X \tag{9}$$

A model having only n constraints in place of (7) and (8) exists for the symmetric m-travelling salesmen problem (see Ali [1]).

Figures 1 and 2
About Here

III. THE ALGORITHM

In this section we present a branch-and-bound algorithm for the m-travelling salesmen problem. The algorithm uses a Lagrangean relaxation, a subgradient algorithm to solve the Lagrangean dual, a greedy algorithm for evaluation of a point of the dual function, penalties to strengthen the lower bounds obtained by the greedy algorithm, and a new concept known as staged optimization. Each of these techniques are explained in the subsections to follow.

3.1 Lagrangean Relaxation

The Lagrangean dual of (6) - (9) selected for this algorithm is as follows:

$$\max_{u,v} \quad \Theta(u,v) \tag{10}$$

where

$$\Theta(u,v) = \min_{x \in X} \{\sum_{ij} c_{ij}x_{ij} + \sum_{i=1}^{n-1} u_i(\sum_{j=1}^{n-1} x_{ij} - 1) + u_n(\sum_{j=1}^{n} x_{nj} - m) + \sum_{j=1}^{n-1} v_j(\sum_{i=1}^{n-1} x_{ij} - 1) + v_n(\sum_{i=1}^{n} x_{in} - m)\}. \tag{11}$$

After rearranging terms $\Theta(u,v) = \min_{x \in X} \{\sum_{ij} \bar{c}_{ij}x_{ij}\} - \alpha$ where $\alpha = \sum_{k=1}^{n-1} (u_k + v_k) - m(u_n + v_n)$ and $\bar{c}_{ij} = c_{ij} + u_i + v_j$. The Lagrangean dual has been used by both Held and Karp [11, 12] and Bazaraa and Goode [3] in their highly successful work on the travelling salesman problem. Our relaxation is a natural extension of their model. Lagrangean relaxation for general integer programs has been extensively studied by Geoffrion [10].

Consider the following results which relate the Lagrangean dual and the primal.

Theorem 1. (Bazaraa and Shetty [4])

Let x^* solve (6) - (9) and let (u^*,v^*) solve (10), (11). Then $\Theta(u^*,v^*) \leq \sum_{ij} c_{ij}x^*_{ij}$.

Theorem 2. (Bazaraa and Shetty [4])

$\Theta(u,v)$ is concave.

Theorem 3. (Geoffrion [8])

Let $(\bar{u},\bar{v})$ be any vectors, let $\bar{x} \varepsilon X$ solve $\Theta(\bar{u},\bar{v})$ and let

$$r_i = \begin{cases} \sum_j \bar{x}_{ij} - 1, \text{ for } i = 1, \ldots, n-1 \\ \sum_j \bar{x}_{ij} - m, \text{ for } i = n \end{cases}$$

$$s_j = \begin{cases} \sum_i \bar{x}_{ij} - 1, \text{ for } j = 1, \ldots, n-1 \\ \sum_i \bar{x}_{ij} - m, \text{ for } j - n. \end{cases}$$

The vector (r,s) is a subgradient of $\Theta(u,v)$ at the point $(\bar{u},\bar{v})$.

The above three results are well-known and are easily proved. Given the optimal vectors associated with both the primal and dual, the non-negative quantity

$$\frac{\sum_{ij} c_{ij}x^*_{ij} - \Theta(u^*,v^*)}{\sum_{ij} c_{ij}x^*_{ij}} \qquad (12)$$

is called the *duality gap*.

For this work, the subgradient algorithm ([13, 15]) is used to solve the dual (10) and (11). Our implementation of this general method is presented below.

ALG-1: SUBGRADIENT OPTIMIZATION METHOD FOR DUAL

0. *Parameter Selection and Initialization*

   a. [Select Step Size Parameters] Select $\bar{G}$, $\bar{H}$, $\lambda_i, \ldots, \lambda_{\bar{G}}$.

   b. [Set Initial Solution] $u \leftarrow 0$, $v \leftarrow 0$, $u^* \leftarrow 0$, $v^* \leftarrow 0$, $r^* \leftarrow 0$, $s^* \leftarrow 0$.

   c. [Set Lower Bound] $L \leftarrow -\infty$.

   d. [Obtain Over Estimate] Set $\bar{\Theta}$ such that $\bar{\Theta} \geq \max_{u,v} \Theta(u,v)$.

   e. [Initialize Counters] $i \leftarrow 1$, $h \leftarrow 0$.

1. *Solve Subproblem*

   a. [Evaluate Dual At (u,v)] Let x^* solve $\min_{x \in X} \{\sum_{ij} \bar{c}_{ij} x_{ij}\}$ and set $\Theta \leftarrow \sum_{ij} \bar{c}_{ij} x^*_{ij} - \alpha$.

   b. [Determine Subgradient]

$$\text{Set} \quad r_i \leftarrow \begin{cases} \sum_j x^*_{ij} - 1, \text{ for } i = 1, \ldots, n-1, \\ \sum_j x^*_{ij} - m, \text{ for } i = n \end{cases}$$

$$s_j \leftarrow \begin{cases} \sum_i x^*_{ij} - 1, \text{ for } j = 1, \ldots, n-1, \\ \sum_i x^*_{ij} - m, \text{ for } j = n \end{cases}$$

   c. [Test For Optimality] If $r = s = 0$, stop with (u,v) an optimum for the dual and x^* an optimum for the primal.

   d. [Improved Solution?] If $\Theta > L$, then $L \leftarrow \Theta$, $u^* \leftarrow u$, $v^* \leftarrow v$, $r^* \leftarrow r$, $s^* \leftarrow s$, and go to 2.

e. [Test Step Size Counter] $h \leftarrow h+1$. If $h = \bar{H}$, go to 3.

2. *Move To New Point*

$(u,v) \leftarrow (u,v) + [\lambda_i(\bar{\Theta} - \Theta)/||(r,s)||]\ (r,s)$, and go to 1.

3. *Change Step Size Or Terminate*

If $i = \bar{G}$, terminate with (u^*,v^*) as an optimum for the dual; otherwise, $h \leftarrow 0$, $i \leftarrow i+1$, $(u,v) \leftarrow (u^*,v^*) + \{\lambda_i[\bar{\Theta} - \Theta(u^*,v^*)]/||(r^*,s^*)||\}(r^*,s^*)$, and go to 1.

An excellent discussion of convergence results for the subgradient algorithm are given in Helgason [14].

3.2 Matroidal Structure of M-Trees

To implement *ALG-1* efficiently, one needs a fast procedure for solving

$$\min_{x \in X} \{\sum_{ij} \bar{c}_{ij} x_{ij}\}. \qquad (13)$$

In this section we show that X has a matroidal structure and (13) may be solved by a greedy algorithm.

We now present results from matroid theory which will be used in the development of an efficient algorithm for (13). Recall that a *matroid*, $M = [Z, \psi]$, is a finite set Z and a set ψ of subsets of Z such that the following axioms hold:

M1. $\phi \in \psi$.

M2. If $X \in \psi$ and $Y \subseteq X$, then $Y \in \psi$.

M3. If $U, V \in \psi$ with $|U| = |V| + 1$, then there exists an $x \in U - V$ such that $V \cup \{x\} \in \psi$.

Let 2^Z denote the power set of Z. Then the elements of $\psi \subset 2^Z$ are called *independent* subsets of 2^Z and $2^Z - \psi$ are called *dependent* subsets. A maximal independent subset is called a *base*.

We now show that a greedy algorithm can be used to find a minimum weight base. Let $\beta \subset \psi$ denote the set of bases for some matroid. Let $w : Z \to R$ be a weight function and extend this to $w : 2^Z \to R$ as follows:

$$w(C) = \sum_{e \in C} w(e), \; C \subset Z.$$

The minimal weight base problem may be described as follows:

Find $X \in \beta$ such that

$$w(X) = \min_{B \in \beta} w(B). \qquad (14)$$

An optimal base X may be obtained by employing the following algorithm.

ALG-2: GREEDY ALGORITHM FOR A MINIMAL WEIGHT BASE

0. $X_0 \leftarrow \phi$, $i \leftarrow 1$, $Y \leftarrow Z$.
1. Find x_i such that $w(x_i) = \min\{w(x) : x \in Y, \{x\} \cup X_{i-1} \in \psi\}$. If no such x_i exists, stop, X_{i-1} is an optimum.
2. $X_i \leftarrow X_{i-1} \cup \{x_i\}$, $Y \leftarrow Y - \{x_i\}$, $i \leftarrow i+1$ and go to 1.

Edmonds [6] proved that *ALG-2* produces an optimum for (14).

To apply Edmonds result to problem (13) we must show that X (the set of m-trees) is the set of bases for some matroid. If this can be shown, then (13) can be solved by the efficient greedy algorithm (*ALG-2*).

Let $\hat{A} = \{(i,j) : 1 \leq i \leq n-1, i \leq j \leq n-1, i \neq j\}$, $\tilde{A} = \{(i,n), (n,i) : 1 \leq i \leq n-1\}$, and let $A = \hat{A} \cup \tilde{A}$. Then we will call a set $X \subseteq \hat{A}$ independent if the edges of X do not form a cycle. We will call a set $X \subseteq \tilde{A}$ independent if $|X| \leq 2m$. Furthermore, $X \subset A$ will

be called independent if $|X \cap \hat{A}| \leq n + m - 1$ and $X \cap \hat{A}$ and $X \cap \tilde{A}$ are independent. Let ψ be the set of all independent subsets of A. Clearly a maximal independent subset of A is an m-tree. Hence, we need only show that $[A, \psi]$ is a matroid to prove that *ALG-2* solves (13).

Before proving that $[A, \psi]$ is a matroid we develop the following preliminary result.

Theorem 4.

Let G be a graph and let Ω be the set of all spanning trees of G. Let $E_1 \subseteq S_1$ and $E_2 \subset S_2$ where S_1, $S_2 \in \Omega$ and $|E_1| = m_1$, $|E_2| = m_2$, $m_1 > m_2$. Then there exists an arc $e \in E_1 - E_2$ such that $E_2 \cup \{e\} \subseteq S_3 \in \Omega$.

Proof. Suppose G has n nodes. Then E_1 has $n - m_1$ components and E_2 has $n - m_2$ components. Suppose E_1 has no arc $e = (i,j)$ such that i and j are in different components of E_2. This implies that E_1 has at least $n - m_2$ components. Then $n - m_1 \geq n - m_2 \Rightarrow m_1 \leq m_2$ which contradicts $m_1 > m_2$. Therefore, there exists an $e \in E_1 - E_2$ which connects two components of E_2 and $E_2 \cup \{e\} \subseteq S_3 \in \psi$. This completes the proof of Theorem 4.

Theorem 4 will be used in the proof of the following result.

Theorem 5.

$[A, \psi]$ is a matroid.

Proof. Axioms M1 and M2 hold trivially. Therefore, we must show that M3 holds. Let U_1 and $U_2 \in \psi$ with $|U_i| = u_i$, $|U_i \cap \hat{A}| = t_i$, $|U_i \cap \tilde{A}| = s_i$, $i = 1, 2$ and $u_1 = u_2 + 1$. Let $e \in U_1 - U_2$.

Case 1: $(s_1 > s_2)$. If $s_1 > s_2$, then there exists an arc $e \in [(U_1 \cap \tilde{A}) - U_2]$ for which $U_2 \cup \{e\} \in \psi$.

Case 2: $(s_1 \leq s_2)$. If $s_1 \leq s_2$, then $t_1 \geq t_2 + 1$. Now $(U_1 \cap \hat{A})$ and $(U_2 \cap \hat{A})$ have no cycles. By Theorem 4, there exists an $e \in (U_1 \cap \hat{A}) - (U_2 \cap \hat{A})$ such that $(U_2 \cap \hat{A}) \cup \{e\}$ has no cycles. Therefore $U_2 \cup \{e\} \subset U_3 \in \psi$.

This completes the proof of Theorem 5.

Therefore the minimal m-tree problem,

$$\min_{x \in X} \{\sum_{ij} \bar{c}_{ij} x_{ij}\},$$

is solvable via ALG-2.

3.3 Separation

In [2] it is shown that the number of feasible solutions for an n city asymmetric m-travelling salesmen problem having m salesmen is

$$\binom{n-1}{m} \frac{(n-2)!}{(m-1)!} . \tag{15}$$

The proof of this result is constructive and shows precisely how one may generate an enumeration tree for this problem.

The tree is generated in two phases with phase 1 corresponding to $\binom{n-1}{m}$ in (15) and phase 2 corresponding to $\frac{(n-2)!}{(m-1)!}$. Assume that the tree is constructed from top down and let the single top node correspond to level 0. All nodes at level ℓ in the enumeration tree will have ℓ arcs fixed and the tree has n-1 levels since fixing n-1 arcs uniquely determines an m-tree. The first phase corresponds to the levels 1 through m while the second phase corresponds to levels m+1 through n-1. The two phases of the enumeration tree construction are now given.

Phase 1: At level 0, construct n-m new nodes by fixing arcs (n,1), (n,2), ..., (n,n-m). For the node with fixed arc (n,j), construct n-m+1-j new nodes by fixing arcs (n,j+1), ..., (n,n-m+1). For any node at level ℓ (<m) having arcs

(n,j_1), ..., (n,j_ℓ) fixed where $j_{k+1} > j_k$, construct $n-m+\ell-j_\ell$ new nodes by fixing $(n,j_\ell+1)$, ..., $(n, n-m-\ell)$.

Phase 2: At level $\ell \geq m$, for any partial solution having ℓ arcs fixed, there are $n-1-\ell$ nodes which have no fixed arc into them. Choose one of these nodes, say q. Then $n-\ell+m-2$ new nodes are constructed by fixing the appropriate arcs whose "to" nodes is q. In developing the enumeration tree, the node (city) selected to create the new nodes is the one whose component in the subgradient differs from 0 the most.

A partial enumeration tree for six cities and two salesmen is illustrated in Figure 3.

Figure 3 About Here

3.4 Problem Selection

Following the notational conventions of Geoffrion and Marsten [9], we represent the enumeration tree by a set of candidate problems which are maintained in the candidate list. Let CP_i denote a candidate problem with fixed arcs (i_1, j_1), ..., (i_ℓ, j_ℓ). Define Y_i corresponding to CP_i as $Y_i = \{(x_{11}, \ldots, x_{nn}) : x_{i_1 j_1} = \ldots = x_{i_\ell j_\ell} = 1\}$. Then CP_i is the problem.

$$\min \sum_{ij} c_{ij} x_{ij}$$

$$\sum_k x_{ij} = \begin{cases} 1, & \text{for } j = 1, \ldots, n-1 \\ m, & \text{for } j = n \end{cases}$$

$$\sum_k x_{ik} = \begin{cases} 1, & \text{for } i = 1, \ldots, n-1 \\ m, & \text{for } i=n \end{cases}$$

$$x \in X \cap Y_i.$$

We make use of two relaxations of CP_i in the branch-and-bound algorithm. The relaxation $\overline{CP}_i^1$ is the Lagrangean dual,

$$\max \Theta(u,v)$$

where $\Theta(u,v) = \min\limits_{x\in X\cap Y_i} \{\sum\limits_{ij} \bar{c}_{ij}x_{ij}\} - \alpha$, while $\overline{CP}_i^2$ is simply $\Theta(\bar{u},\bar{v}) = \min\limits_{x\in X\cap Y_i} \{\sum\limits_{ij} \bar{c}_{ij}x_{ij}\} - \alpha$. $\overline{CP}_i^1$ is used only at the initial node in the enumeration tree, and $\overline{CP}_i^2$ is used at all other nodes where $(\bar{u},\bar{v})$ is the optimal solution of $\overline{CP}_i^1$.

Let $v(\overline{CP}_i^2)$ denote the optimal objective value of $\overline{CP}_i^2$. Then a lower bound for all nodes constructed from CP_i is $v(\overline{CP}_i^2)$. Let CP_{i+1} be any descendent of CP_i with $\ell + 1$ fixed arcs and let the arcs selected by the greedy algorithm for $\overline{CP}_i^2$ and $\overline{CP}_{i+1}^2$ be given by (i_1,j_1) $\cdots$ (i_ℓ,j_ℓ) $(\bar{i}_{\ell+1},\bar{j}_{\ell+1})$ $\cdots$ $(\bar{i}_{n+m},\bar{j}_{n+m})$ and (i_1,j_1) $\cdots$ (i_ℓ,j_ℓ) $(\hat{i}_{\ell+1},\hat{j}_{\ell+1})$ $\cdots$ $(\hat{i}_{n+m},\hat{j}_{n+m})$. Since the greedy algorithm was used to obtain the solution to $\overline{CP}_i^2$ and $\overline{CP}_{i+1}^2$, then

$$\bar{c}_{\hat{i}_k\hat{j}_k} \geq \bar{c}_{\bar{i}_{k-1}\bar{j}_{k-1}} \quad \text{for } k = \ell+2, \ldots, n+m. \tag{16}$$

Thus, summing (16) we obtain

$$\sum_{k=\ell+2} \bar{c}_{\hat{i}_k\hat{j}_k} \geq \sum_{k=\ell+2} \bar{c}_{\bar{i}_{k-1}\bar{j}_{k-1}}. \tag{17}$$

Adding $\sum\limits_{k=1}^{\ell} \bar{c}_{i_kj_k} + \bar{c}_{\bar{i}_{m+n}\bar{j}_{m+n}} + \bar{c}_{\hat{i}_{\ell+1}\hat{j}_{\ell+1}}$ to both sides of (17), we obtain

$$v(\overline{CP}_{i+1}^2) \geq v(\overline{CP}_i^2) + \bar{c}_{\hat{i}_{\ell+1}\hat{j}_{\ell+1}} - \bar{c}_{\bar{i}_{m+n}\bar{j}_{m+n}}. \tag{18}$$

Then the right-hand-side of (18) provides a lower bound for CP_{i+1}. The candidate problem selected at each iteration is the one with smallest lower bound.

3.5 The Algorithm

Using the ideas of the previous sections, we now summarize the new branch-and-bound algorithm for the asymmetric m-travelling salesmen problem. The algorithm incorporates a new idea which we call *staged optimization*. Following the conventions of [9], let CL denote the candidate list and INC denote the objective value of the incumbent. Let $\hat{Z}^*$ denote an estimate of the value of the optimal solution. This estimate is based on the observed duality gap and, of course, may be either larger or smaller than the optimal value. The branch-and-bound algorithm is executed with $\hat{Z}^*$ playing the usual role of the incumbent, INC. This may substantially aid the fathoming routine with the risk of fathoming an optimum. If a feasible solution is found such that INC $\leq \hat{Z}^*$, then the fathoming strategy using $\hat{Z}^*$ was justified and the algorithm guarantees optimality. If the complete tree is fathomed without obtaining a feasible solution, then the fathoming strategy was not justified and the optimum has objective value greater than $\hat{Z}^*$. For this case $\hat{Z}^*$ is increased and the procedure is repeated. A similar idea has been reported by Marsten and Morin [16].

The algorithm incorporating staged optimization follows:

ALG-3: BRANCH-AND-BOUND METHOD FOR M-TRAVELLING SALESMEN PROBLEM

0. *Initialization*

   Choose ∂_1, ∂_2, ∂_3, ... for staged optimization, set $t \leftarrow 1$, and set INC $\leftarrow \infty$.

1. *Solve Dual*

   Use *ALG-1* to solve the dual. Let (u^*,v^*) denote the optimal

dual variables and let x^* solve $\min_{x \in X}\{\sum_{ij} \bar{c}_{ij}x_{ij}\}$. Set the lower bound $L \leftarrow \Theta(u^*,v^*)$ and let (r^*,s^*) denote the subgradient of $\Theta(u^*,v^*)$.

2. *Test For Duality Gap*

   If $r^* = s^* = 0$, stop there is no duality gap and x^* solves the primal; otherwise, set $\hat{Z}^* \leftarrow L(1 + \partial_t)$.

3. *Construct First Level In Enumeration Tree*

   Construct the first n-m nodes in the enumeration tree and place them in the candidate list.

4. *Solve New Candidate Problem*

   Let $Y \in CL$ and let x^Y solve $\min_{x \in X \cap Y}\{\sum_{ij} \bar{c}_{ij}x_{ij}\}$. Let $\Theta_Y = \sum_{ij} \bar{c}_{ij} x^Y_{ij}$, and (r_Y, s_Y) denote the subgradient.

5. *Fathom Test*

   If $\Theta_Y > \hat{Z}^*$ go to 7.

6. *Feasibility Test*

   If $r_Y = s_Y = 0$, $x^* \leftarrow x^y$, $INC \leftarrow \Theta_Y$, $\hat{Z}^* \leftarrow \Theta_Y$, fathom candidate problems with lower bounds greater than Θ_Y; otherwise go to 9.

7. *Termination Test*

   If the candidate list is not empty, then go to 4.

8. *New Stage Required?*

   If $INC \neq \hat{Z}^*$, then set $t \leftarrow t + 1$, $L \leftarrow \hat{Z}^*$, and go to 3; otherwise, stop with x^* as the optimum.

9. *Separate*

   Separate Y, update the candidate list, and go to 4.

Note that *ALG-3* can be easily converted into a method to find approximate solutions. The interval of uncertainty at any point in the procedure is [INC, L]. An additional test in step 6 is required to make this conversion.

IV. COMPUTATIONAL EXPERIENCE

The branch-and-bound procedure, *ALG-3*, has been coded in standard FORTRAN for an in-core implementation. The code was initially tested on randomly generated asymmetric problems on a CDC 6600. The random number generator employed for the generation of problems is the one available on the CDC FTN compiler. The range used for generating the distances was [100, 3400]. The same range was used by Bazaraa and Goode in their computational work on the travelling salesman problem [3]. Since the code is designed for an in-core implementation, one of the major problems encountered in the computational testing was core storage. The size of the candidate list grows rapidly for the larger problems, $n > 50$ and the storage requirement soon exceeds the available 200K octal words, even with most of the data packed.

The computational efficiency of the code is sensitive to the selection of parameters. Bazaraa and Goode observed that there is a trade-off in the amount of computational effort expended in the maximization of the dual and the branch-and-bound procedure. The more time spent in *ALG-1*, the better the lower bound obtained. We employed $\lambda_i = 2^{-i}$ in the code and $\bar{G}$ and $\bar{H}$ were selected based on the size of the problem. The subgradient procedure increases the lower bound L rapidly in the initial iterations and minimal increases are obtained for $\bar{G} > 10$. However, for larger problems, we found it beneficial to use a larger value for $\bar{G}$, even though the relative increase in the lower bound is minimal. Further, the choice of the duality gap estimates ∂_i is crucial. If they are chosen too large, then the candidate list grows rapidly, while if they are too small,

the number of candidate problems solved increases.

Our initial computational experience centered on evaluating the behavior of the code to parameter selection and to the type of problem being solved. Table 1 summarizes computational results for asymmetric problems for n = 30, 40, 50, and 60 with $1 \leq m \leq n/10$. The parameter settings for the first seven sets used were $\bar{H} = 5$, $\bar{G} = 10$ and $\partial_1 = .01$, $\partial_2 = .02$, and $\partial_3 = .03$. We found that the duality gaps for problems with n = 50 and 60 were smaller than .005 and further, the number of subproblems generated with lower bounds within one percent of the solution for the Lagrangean dual was large. Problem sets 9 - 12 were solved with the duality gap estimates set to $\partial_1 = .005$, $\partial_2 = .01$, $\partial_3 = .015$.

The general conclusions which may be drawn from the results of problem sets 1 - 12 is that the duality gap seems to decrease as m increases. Further, the problems with larger values of m are easier to solve than those with smaller values of m. However, as the problem size increases, the number of candidate problems within small percentages of the lower bound obtained from the Lagrangean dual increases. To circumvent the consequential storage problem, we have devised an approximation technique which simply consists of terminating the branch-and-bound procedure as soon as a solution is obtained. The solution so obtained is provable to be within a small percentage of the optimal solution. Since the staged optimization technique makes use of estimates of the duality gap of a problem, it is possible to obtain a suboptimal solution within a known percentage of the true solution by terminating the branch-and-bound procedure as soon as an m-tour is obtained. This approximate solution technique was tested on problems for n = 60. The results are summarized

in problem sets 13 - 18 of Table 1. To obtain a tighter lower bound, $\bar{G} = 12$ was used with $\partial_1 = .005$, $\partial_2 = .01$, and $\partial_3 = .02$.

Table 1 About Here

Table 2 reports computational results for the solution of 25 100-city problems for which solutions were obtained. There were 25 other problems which terminated due to storage limitations before locating an m-tour. For the use of the approximate technique, a better selection procedure, which makes use of the subgradient, can be devised so that an m-tour can be located before the number of candidate problems becomes unmanageable (see [1]).

Table 2 About Here

To gain insight into the solution of symmetric m-travelling salesmen problems, we attempted to solve 50 100-city problems by randomly generating such problems. The range on the costs used was the same as for asymmetric problems. For such problems, the code was slightly modified so that the subgradient employed had n rather than 2n components. However, the enumeration tree was not modified for these problems, nor were other specializations made. Computational results for 15 problems which were successfully solved are given in Table 3. The remaining 35 were terminated due to storage limits. The problems which were solved had negligible duality gaps, and thus solutions were obtained before the candidate list grew. The computational results indicate that the duality gap for large problems is exceedingly small. Thus, even though minimal increases are obtained in the course of final iterations of the

subgradient procedure for maximizing the dual, it is beneficial to employ larger values for $\bar{G}$.

Table 3 About Here

Table 4 gives results on Euclidean problems obtained by the use of intercity distances which were provided by the Civil Aeronautics Board for 59 cities. Because the code has not been designed for the solution of such problems specifically, the computational experience on these problems is not extensive. Rather, the focus here was to obtain insight into the nature of solutions to such problems. Five networks were chosen arbitrarily and three problems were defined for each network. The first four networks have 30 nodes each. Networks III and IV differ only in the choice of the base node. The solutions to the problem are illustrated in Figures 4 - 8. The most interesting inference that may be drawn from the solutions is the relationship between the duality gaps for the problems and the solutions. The solutions to the six problems on networks I and II are intuitively obvious, whereas the solutions to the problems on networks III and IV are not. Thus, the harder the problem, the larger the duality gap. Furthermore, note the manner in which solutions to problems on the same network are related. From the first three networks, it would seem that realistic m-travelling salesmen problems would have to be further constrained to ensure that each salesman visit at least some minimum number of cities. Failing this constraint, solutions of the kind illustrated may be expected. Whereas, as the number of salesmen increases, as in Figure 7, the solutions become more meaningful. That is, not all but

one salesmen travel to only one other city. The solutions obtained for the network on 59 nodes are illustrated in Figure 8. For the problems on networks III, IV and V, even though the initial estimates of the upper bounds used were quite close to the true duality gap, the number of subproblems examined before verification of optimality is large.

Table 4 About Here

Figures 4 - 8 About Here

Based on our computational experience using the technology developed in this exposition, we have arrived at the following conclusions:

(i) The duality gap for problems with up to 100 cities is quite small (less than 1%).

(ii) Holding the number of cities constant, the problems become easier to solve as the number of salesmen increases and the duality gap decreases.

(iii) Exact solutions for problems with up to 50 cities can be obtained routinely with the implementation of the algorithm reported.

(iv) A in-core/out-of-core algorithm using this technique could be used to solve problems having 100 cities.

REFERENCES

1. Ali, A. I., "Two Node-Routing Problems," unpublished dissertation, Operations Research Department, Southern Methodist University, Dallas, Texas (1980).
2. Ali, A., and J. Kennington, "The M-Travelling Salesmen Problem," Technical Report OR 80016, Department of Operations Research, Southern Methodist University, Dallas, Texas (July 1980).
3. Bazaraa, M. S., and J. J. Goode, "The Travelling Salesman Problem: A Duality Approach," Mathematical Programming, 13, (1977), 221-237.
4. Bazaraa, M. S., and C. M. Shetty, Nonlinear Programming: Theory and Algorithms, John Wiley and Sons, New York, NY, (1979).
5. Bellmore, M., and S. Hong, "Transformation of Multi-Salesmen Problem to the Standard Travelling Salesmen Problem," Journal of the Association of Computing Machinery, 21, 3, (1974) 500-504.
6. Edmonds, J., "Matroids and the Greedy Algorithm," Mathematical Programming, 1, (1971), 127-136.
7. Gavish, B., and K. Srikanth, "Optimal and Sub-Optimal Solution Methods for the Multiple Travelling Salesman Problem," Presented at the Joint National TIMS/ORSA Meeting, Washington, D. C. (1980).
8. Geoffrion, A. M., "Duality in Nonlinear Programming: A Simplified Applications-Oriented Development," SIAM Review, 13, 1, (1971) 1-37.
9. Geoffrion, A. M., and R. E. Marsten, "Integer Programming Algorithms: A Framework and State-Of-The-Art Survey," Management Science, 18, 9, (1972), 465-491.
10. Geoffrion, A. M., "Lagrangean Relaxation for Integer Programming," Mathematical Programming Study 2, (1974), 82-114.

11. Held, M., and R. M. Karp, "The Travelling Salesman Problem and Minimum Spanning Trees," Operations Research, 18, (1970), 1138-1162.

12. Held, M., and R. M. Karp, "The Travelling Salesman Problem and Minimum Spanning Trees: Part II," Mathematical Programming, 1, (1971), 6-25.

13. Held, M., P. Wolfe, and H. Crowder, "Validation of Subgradient Optimization," Mathematical Programming, 6, (1974), 225-248.

14. Helgason, R. V., "A Lagrangean Relaxation Approach to the Generalized Fixed Charge Multicommodity Network Flow Problem," unpublished dissertation, Department of Operations Research, Southern Methodist University (1979).

15. Kennington, J., and R. Helgason, Algorithms for Network Programming, John Wiley and Sons, New York, NY, (1980).

16. Marsten, R. and T. Morin, "Branch-And-Bound Methods With Stronger Than Optimal Bounds," Presented at Joint National Meeting of TIMS/ORSA in Washington, D. C. (May 1980).

17. Miller, C. E., A. W. Tucker, and R. A. Zemlin, "Integer Programming Formulation of Traveling Salesman Problems," Journal of the Association of Computing Machinery, 7 (1960) 326-329.

18. Svetska, J. A., and V. E. Huckfeldt, "Computational Experience With An M-Salesman Traveling Salesman Algorithm," Management Science, 19, (1973), 790-799.

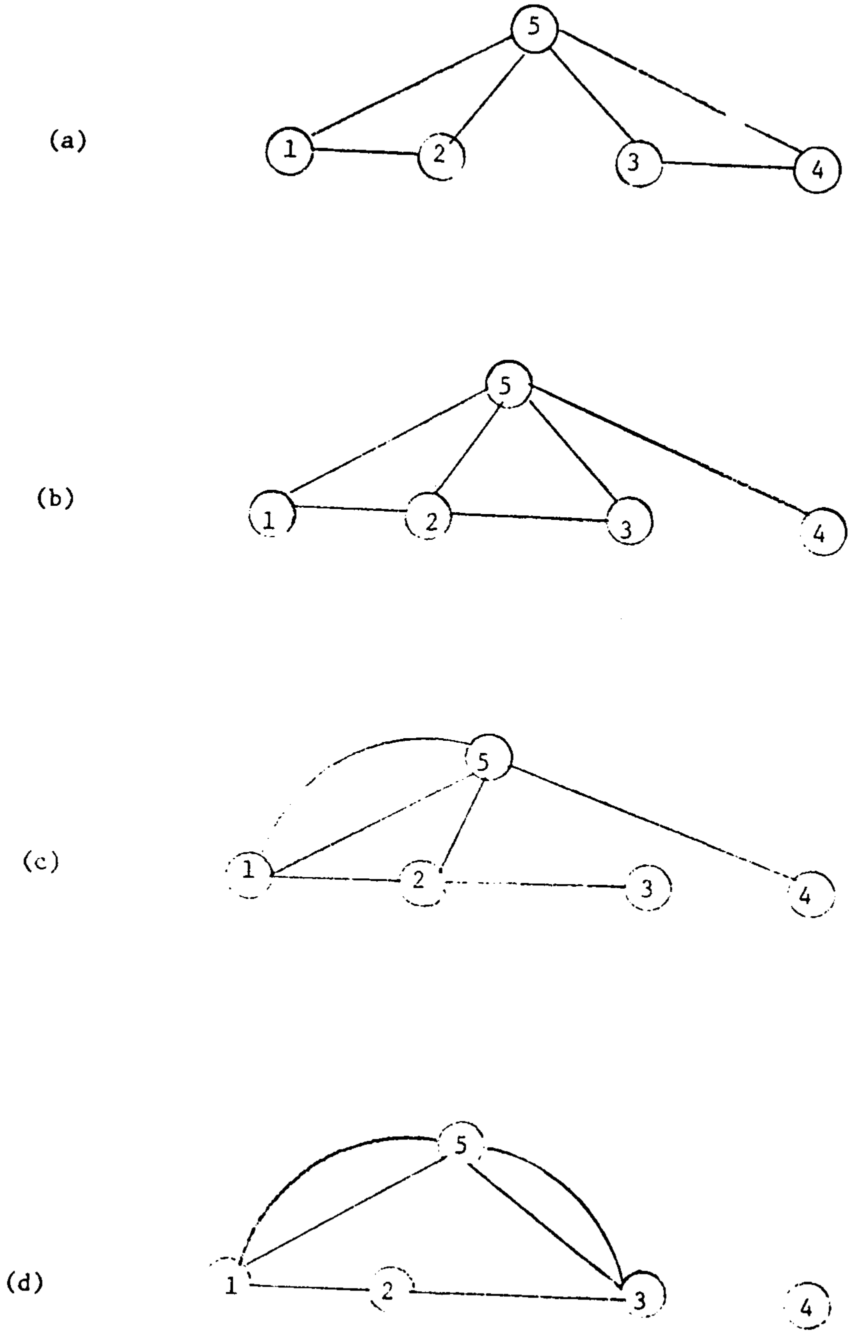


Figure 1. Examples of 2-trees on 5 Nodes

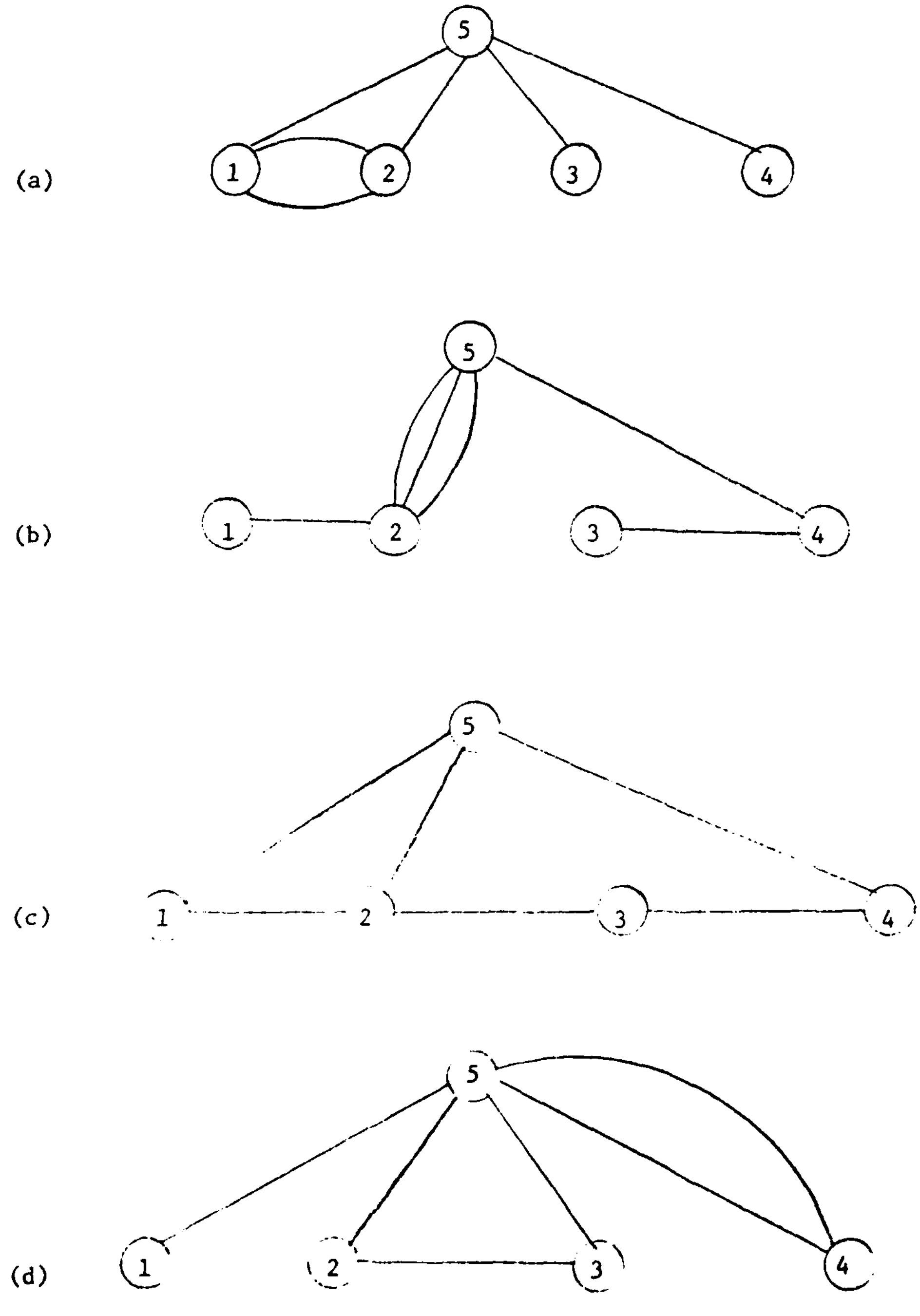


Figure 2. Graphs Having 6 Edges Which Are Not 2-trees

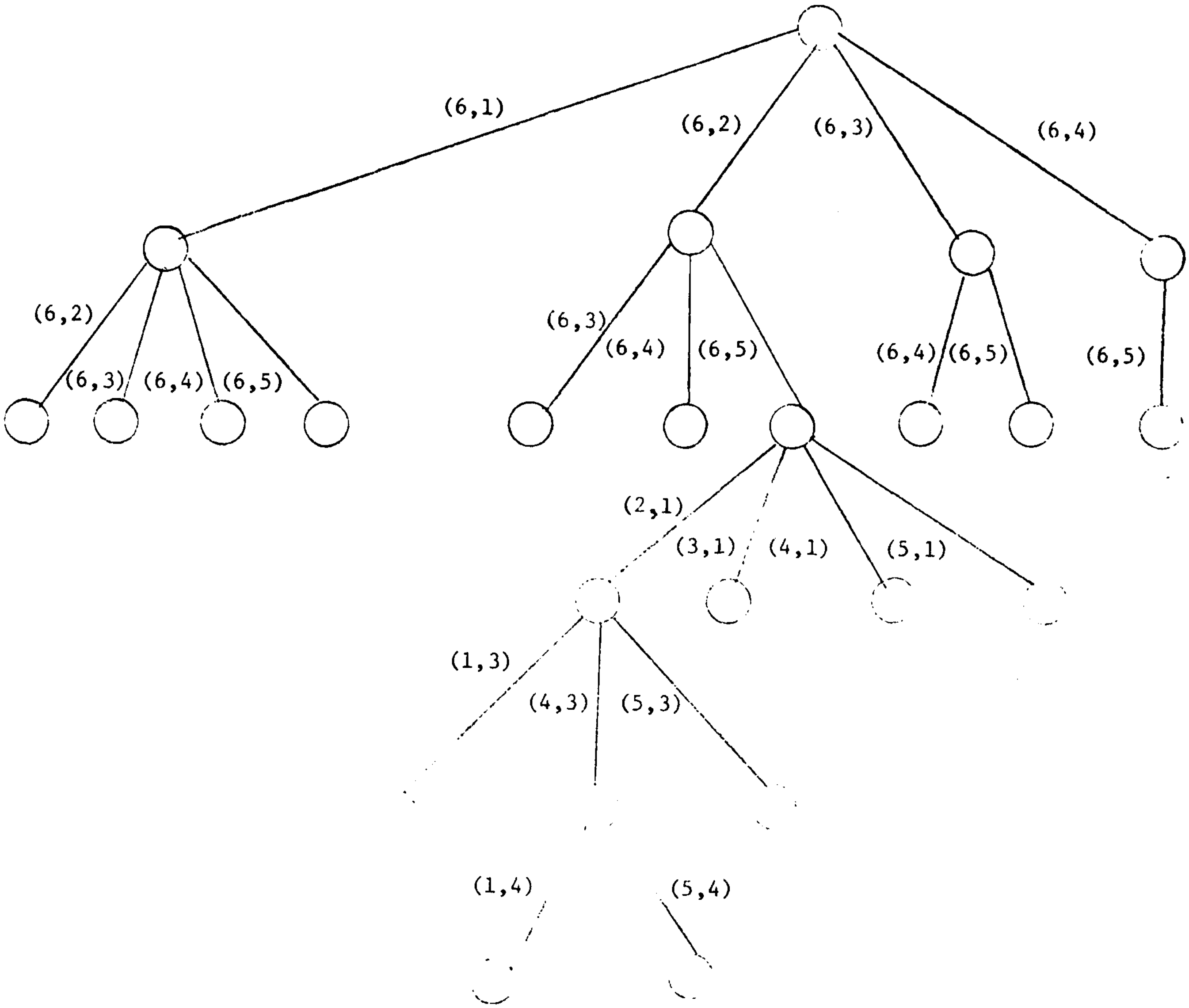
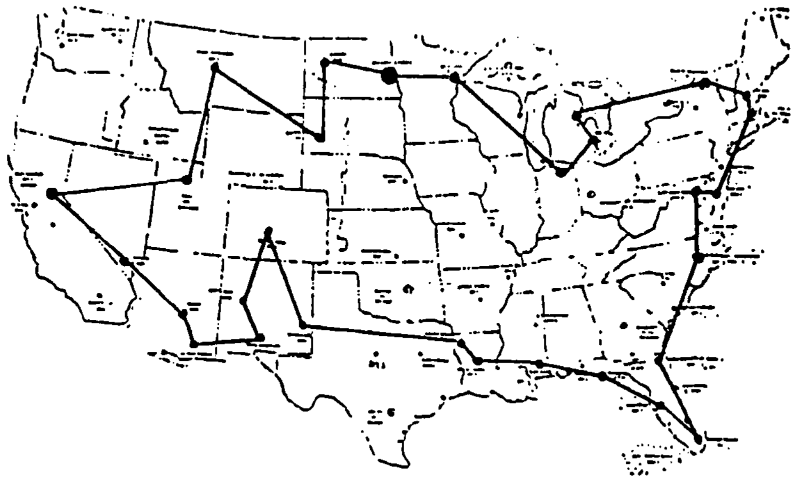
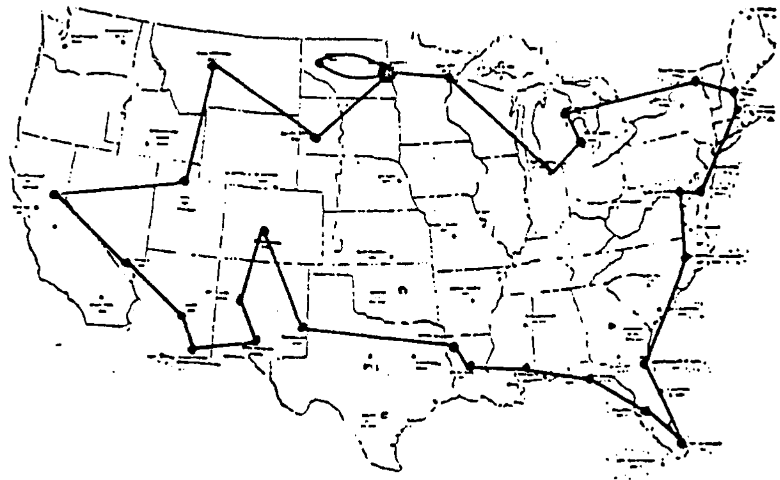


Figure 3. Partial Enumeration Tree For n = 6 m = 2.

(a)



(b)



(c)

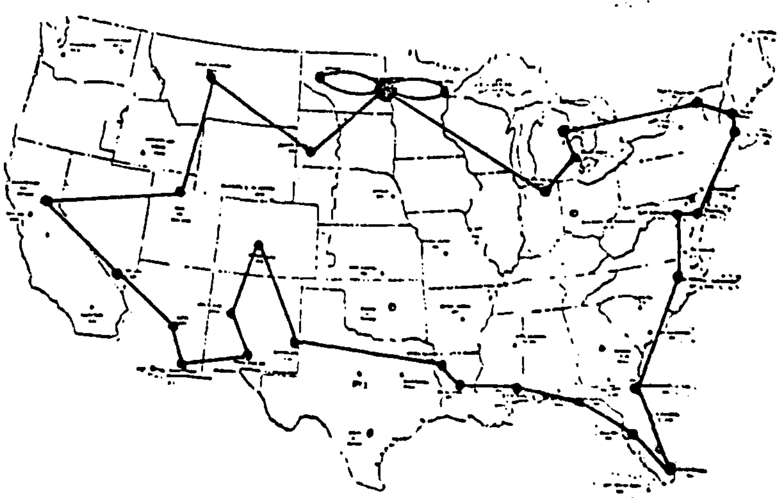


Figure 4. Illustration of Solutions to Problems On Network I.

(a)

(b)

(c)

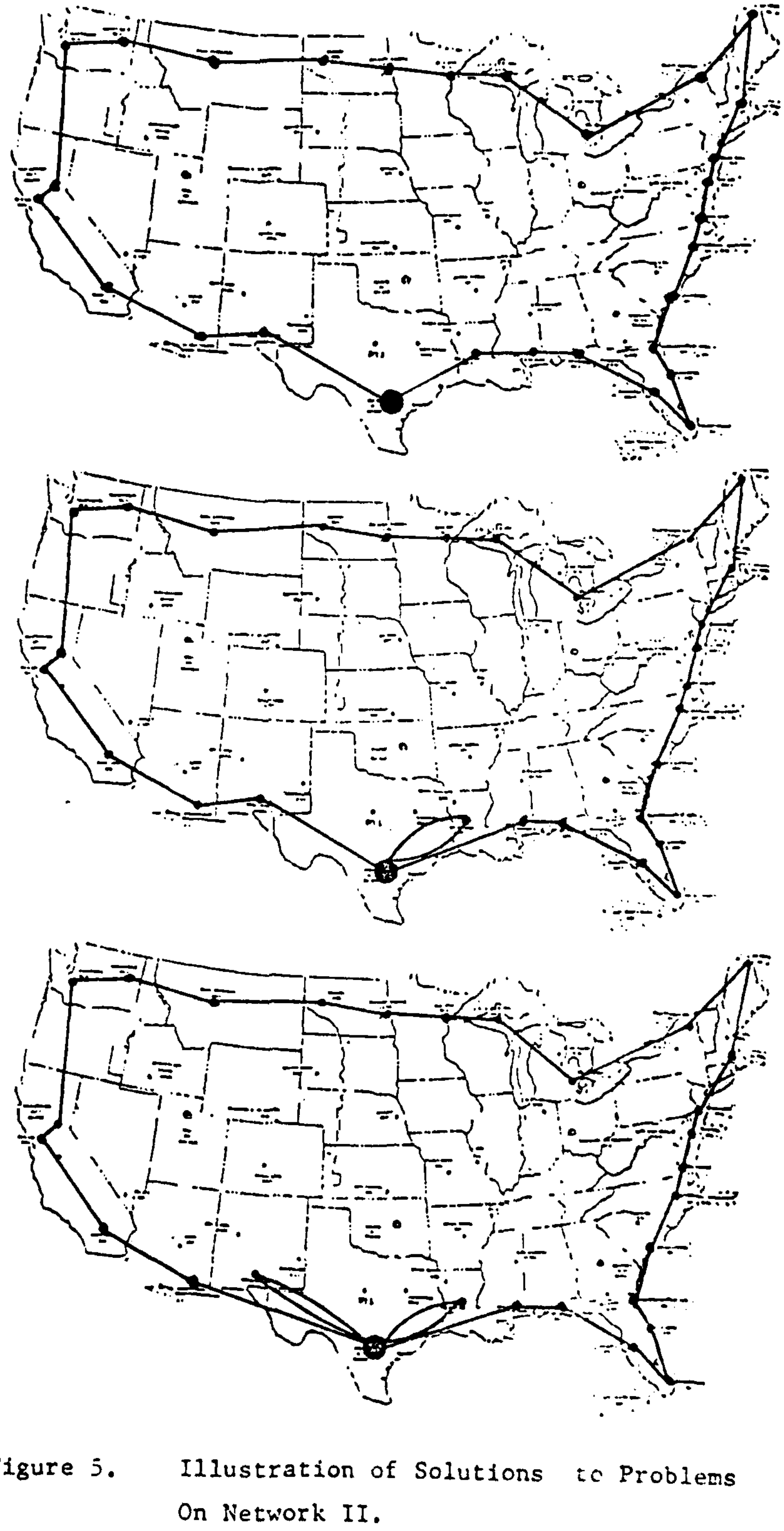


Figure 5. Illustration of Solutions tc Problems On Network II.

(a)

(b)

(c)

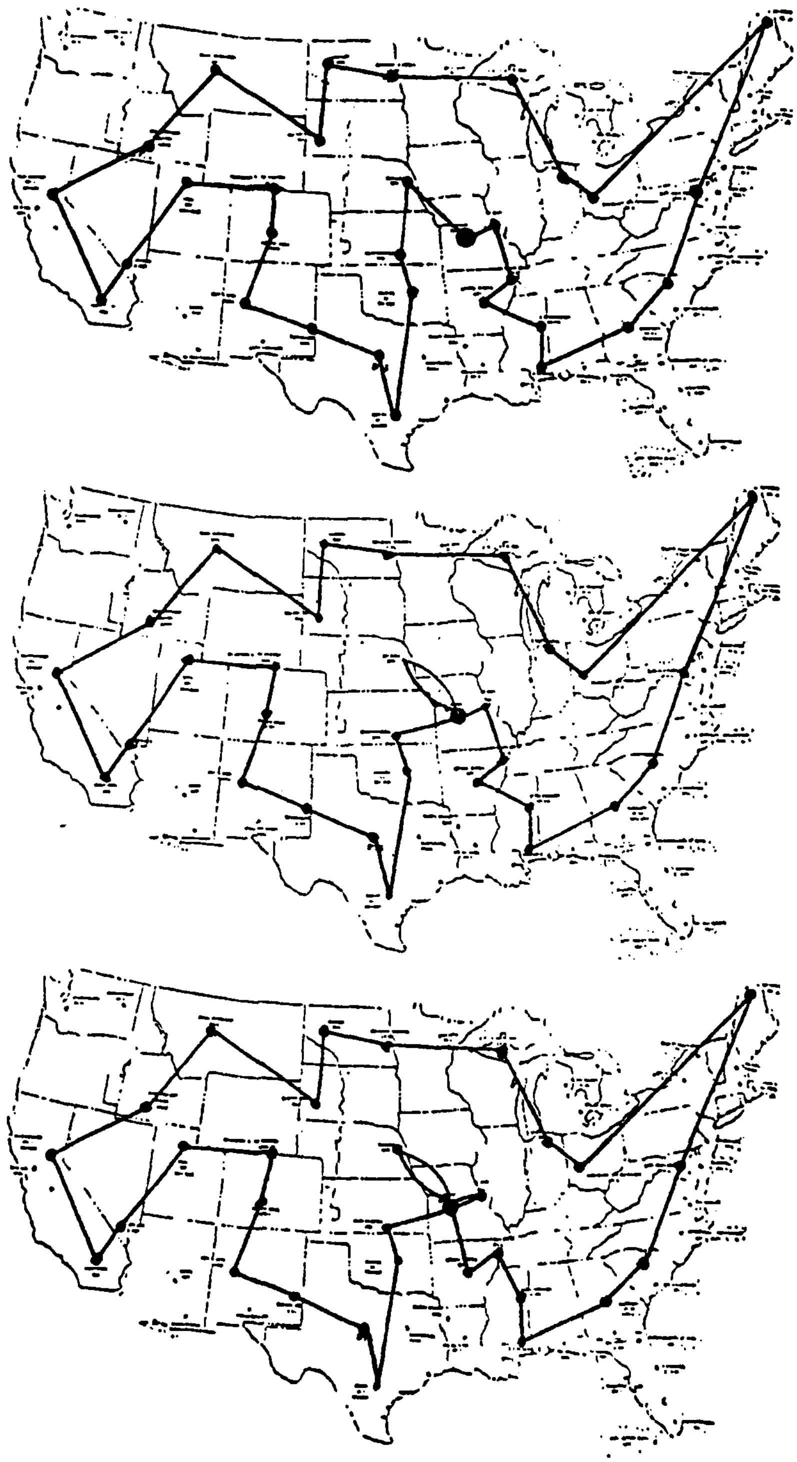
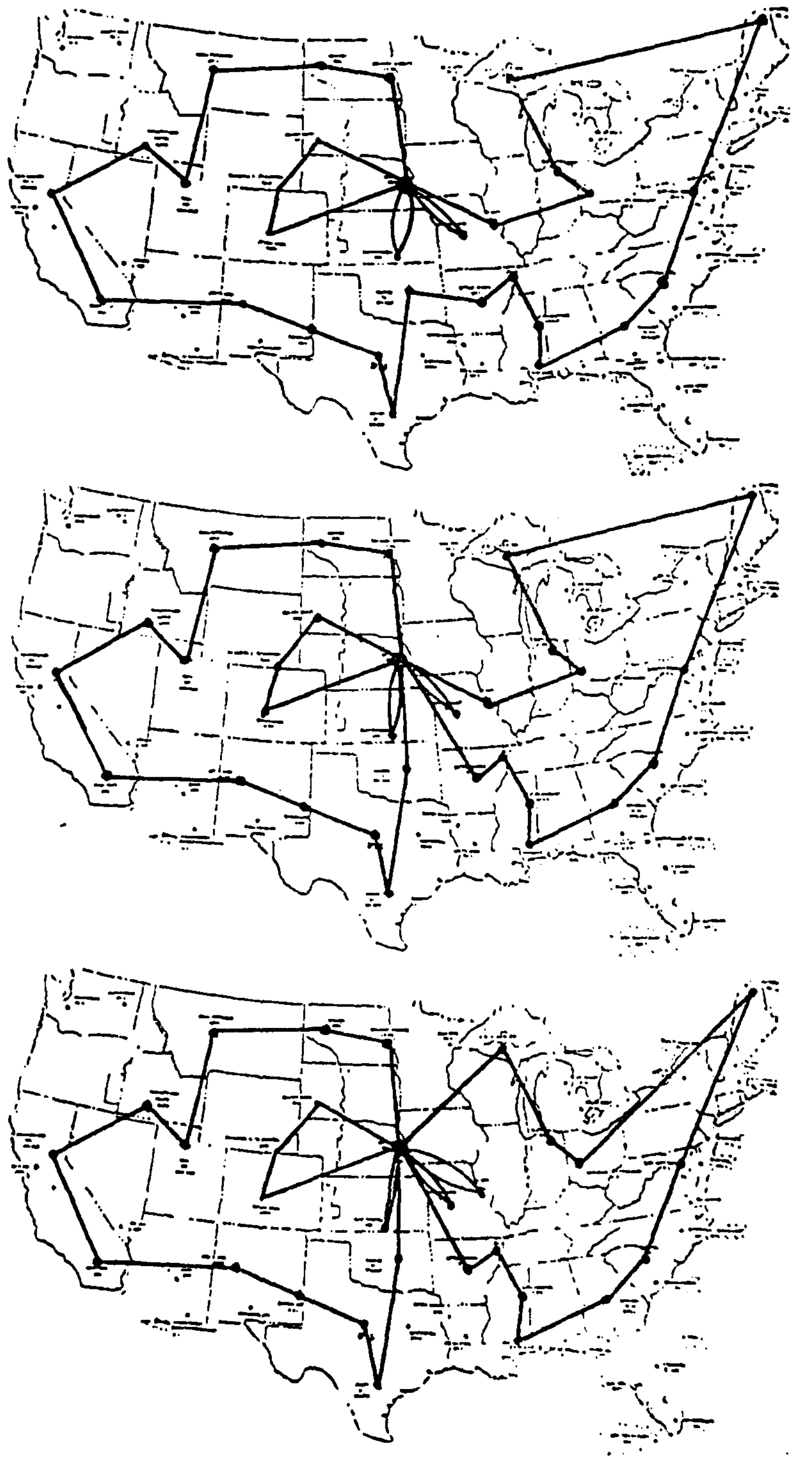


Figure 6. Illustration of Solutions to Problems On Network III.



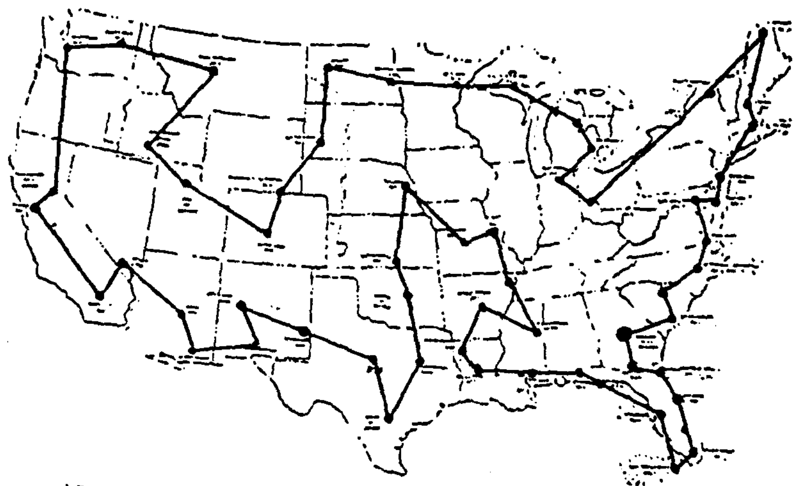
(a)

(b)

(c)

Figure 7. Illustration of Solutions to Problems On Network IV.

(a)



(b)

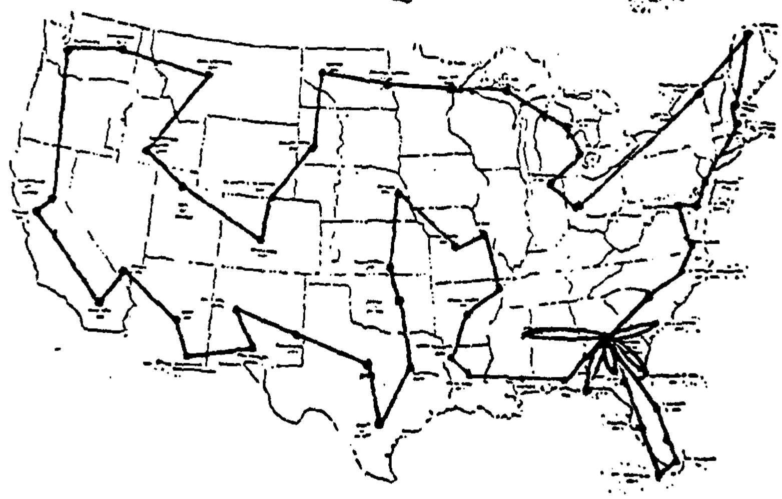


Figure 8. Illustration of Solutions to Problems on Network V.

Table 1. Summary of Computational Results for the Asymmetric m-travelling Salesmen Problem. Each problem set contains 10 problems and all timings are in CPU seconds on a CDC 6600.

Problem Set	n	m	Solution Time for ALG-1	Number of Subproblems	Duality Gap	Solution Time	Average Solution Time
1	30	1	1.9 - 5.4	0 - 2871	0. - .019	1.9 - 40.2	10.7
2	30	2	1.9 - 3.3	0 - 306	0. - .009	1.9 - 5.5	3.8
3	30	3	1.9 - 3.3	0 - 425	0. - .013	1.9 - 6.5	3.5
4	40	1	2.7 - 9.7	0 - 2022	0. - .008	2.7 - 36.5	17.9
5	40	2	5.0 - 7.5	0 - 1955	0. - .012	4.9 - 33.2	12.8
6	40	3	2.7 - 6.0	0 - 541	0. - .005	2.7 - 13.9	6.9
7	40	4	2.7 - 5.2	0 - 875	0. - .003	2.7 - 16.6	6.5
8	50	1	3.8 -16.6	0 - 6134	0. - .007	3.8 - 199.4	52.6
9†	50	2	3.8 -11.1	0 - 1489	0. - .005	3.8 - 39.9	16.4
10	50	3	3.8 - 9.5	0 - 895	0. - .003	3.8 - 28.2	10.6
11	50	4	3.8 - 9.1	0 - 1244	0. - .003	3.8 - 34.4	14.6
12	50	5	3.8 - 8.8	0 - 906	0. - .004	3.8 - 27.7	12.6
13*†	60	1	5.1 -21.7	0 - 6403	0. - .009	5.1 - 223.5	95.8
14*	60	2	5.2 -20.0	0 - 2651	0. - .009	5.1 - 90.9	36.8
15*†	60	3	5.1 -16.9	0 - 296	0. - .009	5.1 - 23.4	11.7
16*	60	4	5.1 -14.8	0 - 2130	0. - .009	5.1 - 67.9	20.8
17*	60	5	5.1 -15.5	0 - 384	0. - .009	5.1 - 25.4	14.3
18*	60	6	5.1 -15.0	0 - 2096	0. - .008	5.1 - 67.8	19.8

* Results are for Suboptimal Solutions. (Interval of Uncertainty = Duality Gap)
† Some problems in the set were not solved due to storage and time limits. (For set 9, 1; for set 13; 2, for set 15, 1).

Table 2. Approximate Solutions For the Asymmetric m-travelling Salesmen Problem on 100 cities. All timings are in CPU seconds on a CDC 6600 and the interval of uncertainty = Duality Gap.

Problem No.	Problem Seed	m	Lagrangean Dual			Branch and Bound		Duality Gap	Total Time
			$\Theta(0,0)$	$\Theta(u^*,v^*)$	Time	Solution	Nodes		
1	368	2	12255	15478	56.	15512	5660	.0021	471.
2	368	5	13784	17122.7	50.	17208	411	.0049	83.
3	368	7	15128	18554	44.	18620	255	.0035	64.
4	49	1	12245	15901.9	90.	15965	519	.0039	125.
5*	49	5	13733	17437.9	50.	17563	16	.0070	51.
6	1763	1	12244	15720.9	74.	15791	381	.0044	109.
7	478	3	12583	15687	12.9	15687	0	.0000	13.
8	478	6	14053	17337	41.5	17356	102	.0010	50.
9	391	3	12678	16498	13.0	16498	0	.0000	13.
10	5513	5	13029	16811.9	48.6	16827	720	.0008	104.
11	3801	3	12580	15853.9	54.	15625	76	.004	60.
12	3801	5	13384	16781.5	51.	16869	1729	.0052	175.
13	3801	8	15432	18989	13.	18989	0	.0000	13.
14*	3801	10	17131	20656.2	42.9	20809	2144	.0070	201.

Parameters: $\overline{H} = 5$, $\overline{G} = 15$, $\partial_1 = .005$, $\partial_2 = .01$, $\partial_3 = .02$

* $\partial_1 = .01$

Table 2 continued

Problem No.	Problem Seed	m	Lagrangean Dual			Branch and Bound		Duality Gap	Total Time
			$\Theta(0,0)$	$\Theta(u^*,v^*)$	Time	Solution	Nodes		
15	1501	1	12058	15568.5	76.7	15628	1938	.003	243.
16	1501	3	12523	16237.1	51.	16276	441	.002	84.
17	1501	8	15393	19393	13.	19319	0	0.0	13.
18	4203	1	12134	15149.7	66.	15184	4397	.002	379.
19*	4203	3	12459	15491	52.	15607	5537	.007	478.
20	4203	5	13182	16230.5	50.3	16248	207	.001	68.
21	4203	8	14591	17659.6	51.5	17716	454	.003	87.
22	5513	5	13029	16811.9	48.6	16827	720	.0008	104.
23	1212	3	12561	16181	13.	16181	0	0.0	13.
24	1212	5	13206	16995	13.	16995	0	0.0	13.
25	1212	8	14956	19031.9	43.1	19172	68	.007	49.

Table 3. Approximate Solutions For the Symmetric m-travelling salesmen problem on 100 cities. All timings are in CPU seconds on a CDC 6600 and the interval of uncertainty = Duality Gap.

Problem No.	Problem Seed	m	Lagrangean Dual			Branch and Bound		Duality Gap	Total Time
			$\Theta(0,0)$	$\Theta(u^*,v^*)$	Time	Solution	Nodes		
1	98	1	14069	16771	84.2	16771	1	.0000	84.5
2	837	9	17443	21972	66.5	21972	1	.0000	66.8
3	5078	2	14486	16452.9	86.7	16453	3	.0000	87.5
4	5078	4	15364	17499	73.7	17499	1	.0000	73.9
5	1212	2	14058	16795.9	83.8	16796	1	.0000	84.1
6	1212	7	15191	17955.9	113.	17956	1	.0000	113.7
7	5513	7	16623	20375.9	66.8	20376	1	.0000	67.9
8	3068	6	15664	18491	73.7	18491	1	.0000	74.
9	1045	5	15138	17824.7	87.2	17853	51	.0015	97.1
10	6190	5	15058	18293.8	66.2	18294	7	.0000	67.2
11	492	8	15868	19156.3	77.	19190	26	.0017	80.3
12	394	6	15210	19217.8	83.7	19218	1	.0000	84.
13	606	5	14317	17204.8	73.2	17205	10	.0000	74.8
14	367	6	15469	18674.8	60.	18675	10	.0000	61.9
15	692	1	14569	17463.9	89.3	17464	1	.0000	89.7

Parameters: $\overline{H} = 5$, $\overline{G} = 15$, $\partial_1 = .005$, $\partial_2 = .01$, $\partial_3 = .015$

Table 4. Solution of Euclidean m-travelling Salesmen Problems. All timings are in CPU seconds on a CDC 6600.

Network	$\overline{I}$	m	Lagrangean Dual			Branch and Bound		Duality Gap	Total Time	Figure
			$\Theta(0,0)$	$\Theta(u^*,v^*)$	Time	Optimum	Nodes			
I	30	1	7651	8441	10.6	8441	12	.0000	11.0	4 (a)
I	30	2	7650	8699	10.5	8699	21	.0000	11.1	4 (b)
I	30	3	7951	9182	10.1	9182	8	.0000	10.4	4 (c)
II	30	1	7580	7999	8.3	7999	8	.0000	8.6	5 (a)
II	30	2	8004	8744	6.1	8744	7	.0000	6.4	5 (b)
II	30	3	8712	9723	6.6	9723	8	.0000	6.9	5 (c)
III*	30	1	8077	9724.5	12.8	9806	22136	.0083	281.7	6 (a)
III*	30	2	7768	9895.5	9.72	9977	25682	.0082	301.4	6 (b)
III*	30	3	7772	10170	7.8	10225	7179	.0050	89.8	6 (c)
IV	30	4	8450	11133	8.6	11185	31142	.0040	343.7	7 (a)
IV	30	5	8903	11729.5	7.5	11762	16705	.0020	187.0	7 (b)
IV	30	6	9443	12367	6.6	12386	4419	.0010	54.5	7 (c)
V	59	1	10862	12376	45.3	12423	7761	.0030	494.8	8 (a)
V	59	7	11570	14334.9	48.3	14369	24348	.0023	1314.5	8 (b)

Parameters: $\overline{\Pi} = 5$, $\overline{G} = 20$, $\partial_1 = .005$, $\partial_2 = .01$, $\partial_3 = .015$

* $\partial_1 = .01$

14 CCS-RR-382

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Date Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CCS 382	2. GOVT ACCESSION NO. AD-A092951	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 THE M-TRAVELLING SALESMEN PROBLEM: A DUALITY BASED BRANCH-AND-BOUND ALGORITHM.		5. TYPE OF REPORT & PERIOD COVERED 9 Research rept. 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 I. Ali J. Kennington Southern Methodist University	15	8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-0569 AFOSR-77-3151
9. PERFORMING ORGANIZATION NAME AND ADDRESS Center for Cybernetic Studies, UT Austin Austin, Texas 78712		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research (Code 434) Washington, DC	11	12. REPORT DATE Aug[redacted]80 13. NUMBER OF PAGES 38
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12 40		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

M-travelling salesmen problem, branch and bound, Langrangean relaxation, m-tree, staged optimization, matroid

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper presents a new model and branch-and-bound algorithm for the m-travelling salesmen problem. The algorithm uses a Langrangean relaxation, a subgradient algorithm to solve the Langrangean dual, a greedy algorithm for obtaining minimal m-trees, penalties to strengthen the lower bounds on candidate problems, and a new concept known as staged optimization. Computational experience for both symmetric and asymmetric problems having up to 100 cities is presented.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601

YM

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Date Entered)

406197

DATE
FILMED
8